IN THE CLAIMS:

1-33 (Cancelled)

34. (New) A method for acquiring a lock on a system resourcc in a computer system, the method comprising:

assigning a first memory location to a first system resource, such that a number stored at said first memory location indicates whether said first system resource is locked;

performing an atomic operation that (a) determines if said first memory location contains a first value, indicating that said system resource is not locked and (b) if said first memory location contains said first value, stores a second value, indicating that said first system resource is locked, wherein said atomic opcration is non-interruptiblc; and

if said first memory location does not contain said first value, transferring control to a different operation that will contend for the lock for said first system resource.

35. (New) The method of claim 34, wherein said first memory location is memory that is never swapped out.

36. (New) The method of claim 34, wherein a program or process that initiates said atomic operation can wait for a response from said atomic operation without giving up control of a processor on which it is running.

37. (New) The method of claim 34, wherein said performing step comprises creating a lock message that contains the address of said first memory location and a first number, wherein said first number is added to a second number stored in said first memory location to create a third number and said second number is returned in a response.

38. (New) The method of claim 34, wherein said performing step comprises creating a lock message that contains the address of said first memory location, a first number, and a second number, wherein said first number is compared to a third number stored in

said first memory location and if said first number is equal to said third number, said second number is stored in said first memory location.

39. (New) The method of claim 34, wherein the non-interruptibility of said atomic operation can be modified such that no other operation performed by a channel adapter can intervene, but other host operations can intervene.

40. (New) The method of claim 34, wherein the non-interruptibility of said atomic operation can be modified such that no other operation performed by a first channel adapter performing said atomic operation can intervene, but operations from other channel adapters or other host operations can intervene.

41. (New) The method of claim 34, wherein said first memory location is part of a lock table having a respective memory location for each system resource that can be locked.

42. (New) The method of claim 41, wherein said lock table resides on a single node in a network.

43. (New) The method of claim 41, wherein said lock table is distributed across a plurality of nodes in a network.

44. (New) The method of claim 41, wherein each system resource that can be locked has a respective lock name and a hash on said respective lock name of said first system resource identifies a first node of said plurality of nodes that holds said first memory location.

45. (New) A computer program product for acquiring a lock on a system resource in a computer system, comprising:

a computer readable medium having computer readable program code embodied therein;

computer readable program code configured to assign a first memory location to a first system resource, such that a number stored at said first memory location indicates whether said first system resource is locked;

computer readable program code configured to perform an atomic operation that (a) determines if said first memory location contains a first value, indicating that said system resource is not locked and (b) if said first memory location contains said first valuc. stores a second value, indicating that said first system resource is locked, wherein said atomic operation is non-interruptible; and

computer readable program code configured to transfer control to a different operation that will contend for the lock for said first system resource if said first memory location does not contain said first value.

46. (New) The computer program product of claim 45, wherein said first memory location is memory that is never swapped out.

47. (New) The computer program product of claim 45, wherein a program or process that initiates said atomic operation can wait for a response from said atomic operation without giving up control of a processor on which it is running.

48. (New) The computer program product of claim 45, wherein said atomic operation comprises computer readable program code configured to create a lock message that contains the address of said first memory location and a first number, wherein said first number is added to a second number stored in said first memory location to create a third number and said second number is returned in a response.

49. (New) The computer program product of claim 45, wherein said atomic operation comprises computer readable program code configured to create a lock message that contains the address of said first memory location, a first number, and a second number, wherein said first number is compared to a third number stored in said first memory location and if said first number is equal to said third number, said second number is stored in said first memory location.

50. (New) The computer program product of claim 45, further comprising computer readable program code configured to modify the non-interruptibility of said atomic operation such that no other operation performed by a channel adapter can intervene, but other host operations can intervene.

51. (New) The computer program product of claim 45, further comprising computer readable program code configured to modify the non-interruptibility of said atomic operation such that no other operation performed by any channel adapter can intervene, but other host operations can intervene.

52. (New) The computer program product of claim 45, wherein said computer readable program code is further configured to use a lock table having a respective memory location for each system resource that can be locked.

53. (New) The computer program product of claim 52, wherein said lock table resides on a single node in a network.

54. (New) The computer program product of claim 52, wherein said lock table is distributed across a plurality of nodes in a network.

55. (New) The computer program product of claim 52, wherein each system resource that can be locked has a respective lock name and a hash on said respective lock name of said first system resource identifies a first node of said plurality o nodes that holds said first memory.

56. (New) A computer system, comprising:

a plurality of processors connected such that said plurality of processors share a plurality of system resources;

instructions for assigning a first memory location to a first system resource, such that a number stored at said first memory location indicates whether said first system resource is locked;

instructions for performing an atomic operation that (a) determines if said first memory location contains a first value, indicating that said system resource is not locked and (b) if said first memory location contains said first value, stores a second value, indicating that said first system resource is locked, wherein said atomic operation is non-interruptible; and

instructions for transferring control to a different operation that will contend for the lock for said first system resource if said first memory location does not contain said first value.

57. (New) The computer system of claim 56, wherein said first memory location is memory that is never swapped out.

58. (New) The computer system of claim 56, wherein a program or process that initiates said atomic operation can wait for a response from said atomic operation without giving up control of a processor on which it is running.

59. (New) The computer system of claim 56, wherein said instructions for performing an atomic operation comprise instructions for creating a lock message that contains the address of said first memory location and a first number, wherein said first number is added to a second number stored in said first memory location to create a third number and said second number is returned in a response.

60. (New) The computer system of claim 56, wherein said instructions for performing an atomic operation comprise instructions for creating a lock message that contains the address of said first memory location, a first number, and a second number, wherein said first number is compared to a third number stored in said first memory location and if said first number is equal to said third number, said second number is stored in said first memory location.

61. (New) The computer system of claim 56, further comprising instructions for modifying the non-interruptibility of said atomic operation such that no other operation performed by a channel adapter can intervene, but other host operations can intervene.

62. (New) The computer system of claim 56, further comprising instructions for modifying the non-interruptibility of said atomic operation such that no other operation performed by a channel adapter can intervene, but other host operations can intervene.

63. (New) The computer system of claim 56, wherein said first memory location is part of a lock table having a respective memory location for each system resource that can be locked.

64. (New) The computer system of claim 63, wherein the lock table resides on a single node in a network.

65. (New) The computer system of claim 63, wherein the lock table is distributed across a plurality of nodes in a network.

66. (New) The computer system of claim 63, wherein each system resource that can be locked has a respective lock name and a hash on said respective lock name of said first system resource identifies a first node of said plurality of nodes that holds said first memory location.